

CLOUD STORAGE DENGAN TEKNOLOGI KUBERNETES UNTUK PLATFORM COLLABORATIVE RESEARCH

Muhammad A. Nugroho¹, Rikie Kartadie²

^{1,2} Informatika, STMIK AKAKOM Yogyakarta

Jl. Raya Janti No.143, Jaranan, Bantul, Daerah Istimewa Yogyakarta

e-mail: m.agung.n@akakom.ac.id¹, rikie@akakom.ac.id²

ABSTRAK

Memberikan model alternatif untuk penerapan penelitian kolaboratif melalui layanan teknologi komputasi awan yang diimplementasikan di lingkungan komputer lokal atau untuk menampung penyimpanan file di luar lokasi, penggunaan berbagi file cloud sangat bermanfaat dan mudah. Penggunaan teknologi ini harus diimplementasikan dan diuji keandalannya dalam skala yang baik sehingga jika akan dikembangkan lebih lanjut dapat segera diimplementasikan dan sesuai dengan lingkungan dan sumber daya jaringan yang ada. Penelitian ini berfokus pada implementasi dan pengujian performa kecepatan dan user request yang menghasilkan nilai A pada uji kecepatan pada GTMetrix dan load peak tertinggi connection time 0-5ms. Solusi terhadap penurunan performa dapat disolusikan dengan menggunakan model scaling dikombinasikan dengan proxy, dan load balancing.

Kata Kunci: Collaboration Research, Kubernetes, Penelitian

ABSTRACT

Providing an alternative model for the application of collaborative research through cloud computing technology services implemented in the local computer environment or to accommodate offsite file storage, the use of cloud file sharing is very useful and easy. The use of this technology must be implemented and tested for reliability on a good scale so that if it is to be developed further it can be implemented immediately and by following by under the existing network environment and resources. This research focuses on the implementation and performance testing of speed and user requests that produce an A value on the speed test on GTMetrix and the highest peak load connection time of 0-5ms. A solution to performance degradation can be solved using a scaling model in combination with proxies, and load balancing.

Keywords : Collaboration Research, Kubernetes, Research

I. PENDAHULUAN

Melakukan suatu penelitian yang baik, konsisten, terstruktur selain harus didukung oleh kemampuan metodologi penelitian yang baik juga harus didukung referensi yang relevan, kekinian dan penyimpanan data yang handal. Salah satu kelemahan kita dalam melakukan penelitian yang baik adalah dalam hal manajemen referensi/sitasi dan *file collaboration*. Ketersediaan koneksi internet dan kemajuan teknologi informasi memungkinkan setiap kita dapat melakukan akses pada berbagai sumber referensi. Kemajuan teknologi mobile, memberikan akses mudah terhadap referensi dan dapat dilakukan setiap saat dimanapun kita berada. Aktivitas penelitian kita tidak harus dibatasi oleh lingkungan kampus, perpustakaan dan laboratorium saja namun dimanapun yang bersangkutan terkoneksi dengan internet aktivitas penelitian dapat dilakukan. Istilah *Collaborative research* telah diperkenalkan paling tidak sejak tahun 1986 oleh Beatrice A. Ward, dan Willian J.

Tikunoff [1], dan mereka menyatakan bahwa definisi penelitian kolaboratif disajikan dengan mencatat peran ganda dan kesa-maannya dalam tindakan penelitian. Alasan untuk melakukan penelitian kolaboratif diberikan untuk memusatkan peningkatan penggunaan hasil penelitian, penelitiannya sendiri, dan pengurangan selang waktu dalam pengaplikasian penelitian dan tindakan penelitian. Pada "kecenderungan kolaborasi" masing-masing peneliti, pada sisi sosial berbagai disiplin ilmu akan menunjukkan bahwa kolaborasi adalah fungsi orientasi terhadap tanggung jawab individu versus tanggung jawab kolektif untuk penelitian. Kolaborasi bergantung pada pekerjaan yang dilakukan peneliti — jika kolaborasi bermanfaat, mereka akan melakukannya terlepas dari iklim sosialnya [2].

Program penelitian kolaboratif meningkatkan pentingnya peluang penelitian, hasil dan transfer pengetahuan antara peneliti, universitas dan bisnis yang berpartisipasi. Dengan mulai adanya kesadaran akan pentingnya penelitian kolaboratif dalam mengembangkan kompetensi inti dan dalam membuat keputusan strategis yang tepat, bagaimana cara terbaik menyebarkan sumber daya untuk memenangkan persaingan, kita harus mengembangkan secara terstruktur pendekatan penelitian kolaboratif dan kemudian mengembangkan pengetahuan ini dengan tekad yang sama, namun tentu saja terjadi kesenjangan.

Kerangka kerja kolaboratif dapat menggunakan *file sharing* dan penulisan bersama, tentu saja membutuhkan teknologi yang dapat menjembatani hal ini. Aktivitas penelitian yang baik menuntut adanya proses sharing pengetahuan dengan komunitas peneliti dan kolaborasi aktivitas antar peneliti. Melalui dua hal ini maka setiap peneliti akan selalu terkoneksi dengan sebuah lingkungan penelitian yang kondusif untuk menghasilkan penelitian

yang berkualitas. Melalui *sharing* dan kolaborasi maka keterbatasan fisik lingkungan penelitian yang dihadapi oleh seorang peneliti tidak menjadi kendala untuk menghasilkan sebuah penelitian yang baik dan aplikasi teknologi informasi memungkinkan untuk melakukan upaya *sharing* dan kolaborasi tersebut terlebih di era teknologi awan sekarang ini.

Untuk kepentingan *collaborative research*, NextCloud memiliki fitur yang telah memenuhi, kemudian bagaimana NextCloud diimplementasikan pada lingkungan (*environment*) jaringan yang tersedia dan bagaimana performanya sehingga dapat memenuhi kebutuhan akan *sharing* dan kolaborasi file dalam penelitian. Fokus penelitian ini adalah bagaimana NextCloud dapat diintegrasikan pada *docker cloud environment* dalam skalabilitas teknologi Kubernetes

II. TINJAUAN PUSTAKA

Sebuah tool untuk kepentingan *collaborative research* setidaknya memiliki 4 fungsi utama yaitu: *web-enabled calendar*, *electronic notebook*, *repository tools* dan *database* [3]. *Web-enabled calendar* fungsinya adalah untuk *sharing* agenda dan *milestone research*, sedangkan *electronic notebook* untuk kepentingan *sharing* dokumen dan *tracking research progress*, untuk *repository tools* dan *database* adalah untuk menyimpan segala dokumen terkait dengan pelaksanaan penelitian.

Keuntungan *collaborative research* ini dapat dipenuhi dengan beberapa layanan awan yang telah ada, seperti Google Drive, Amazon S3, Google Kubernetes, Globus dan masih banyak lagi. Salah satu layanan awan adalah NextCloud [4] yang merupakan aplikasi web PHP yang berjalan pada server berbasis linux seperti halnya Apache atau Nginx. NextCloud dapat menyimpan informasi *file sharing*, detail pengguna, data aplikasi, dan konfigurasi sebaik informasi file di MySQL [5].

Server NextCloud ditulis dalam bahasa skrip PHP dan JavaScript. Untuk akses jarak jauh menggunakan *saber/dav*, server WebDAV OpenSource [6]. NextCloud dirancang untuk bekerja dengan beberapa sistem manajemen *database*, termasuk SQLite, MariaDB, MySQL, Oracle Database, dan PostgreSQL [7].

Penelitian yang dilakukan Dewi, L.P., dkk [8] berfokus pada penerapan skalabilitas di kubernetes dan mengevaluasi kinerjanya dalam mengatasi peningkatan jumlah pengguna bersama yang mengakses data akademik. Virtualisasi menggunakan kontainer memungkinkan banyak fleksibilitas untuk manajemen kapasitas di server. Dibeberapa server, skalabilitas server diaktifkan untuk melayani permintaan pengguna. Penerapan skalabilitas ke container (dibeberapa server) mengurangi pod penggunaan CPU karena distribusi beban ke container yang tersebar di banyak pekerja [9].

Dalam tulisan lain, Kariyattin, S, dkk [10], dan Mencomedel, V., dkk., [11] menganalisis *overhead* penerapan dan penghentian container di kubernetes, serta memahami kinerja berbagai konfigurasi pod kubernetes yaitu pengaruh jumlah container per pod. Penelitian yang dilakukan mereka pun menganalisis kluster Kubernetes yang terdiri dari 8 mesin. Model yang diberikan oleh Medel, V, dkk [11] dapat dimanfaatkan sebagai dasar untuk meningkatkan dua kegiatan yaitu: (i)Perencanaan kapasitas dan pengelolaan sumber daya, (ii)Design aplikasi, khususnya bagaimana aplikasi dapat disusun dalam istilah pod dan container. Menurut Medel, V., dkk [11], dapat dilihat bahwa satu container dapat diterapkan dalam interval waktu yang berkisar kurang dari 1 detik hingga 3 detik.

Ada perbedaan dalam performa sistem antara container dan native karena *overhead* virtualisasi. Namun, *overhead* kinerja dalam kontainer dianggap kecil dan dalam beberapa kasus hampir tidak ada [12]. Kinerja setiap server (server FTP, server web, dan server email) yang berjalan pada platform kontainer berbeda. Pengujian ini dilakukan untuk melihat platform terbaik untuk menjalankan server dan dalam kondisi apa hasilnya adalah docker memiliki performa yang lebih lambat dalam total waktu yang dibutuhkan namun stabil dalam penambahan kecepatan.

III. METODE PENELITIAN

Penelitian ini merupakan pengembangan perangkat lunak, dengan sasaran akhir yang diharapkan adalah terbangunnya sebuah perangkat lunak (*software*) sesuai dengan spesifikasi yang diharapkan. Dalam mengembangkan perangkat lunak terdapat banyak paradigma yang bisa digunakan, dan paradigma yang digunakan salah satunya adalah jenis paradigma *prototyping*.

Prototype merupakan sebuah memungkinkan pengembangan untuk membuat model perangkat lunak yang akan direalisasikan. Paradigma *prototyping* pada pengembangan perangkat lunak dapat memungkinkan pengembang untuk membentuk *prototype* awal program. *Prototype* tersebut menjadikan pengembang dapat menyesuaikan rancangan program dengan kebutuhan pemakai dan pemakai mendapatkan gambaran awal mengenai perangkat lunak yang dibuat. Pemakai dapat mengoreksi rancangan yang telah dibuat apakah sudah sesuai dengan yang diharapkan dan menurut M. Beaudouin-lafon and W. E. Mackay [13] desainer harus memutuskan peran apa yang harus dimainkan *prototype* sehubungan dengan sistem akhir dan dalam urutan mana, untuk membuat aspek yang berbeda dari

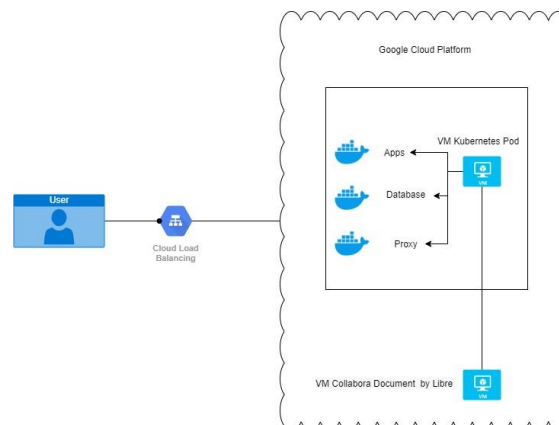
prototype.

Kelebihan paradigma ini adalah terjadi kesesuaian antara permintaan pemakai dengan rancangan yang dibuat oleh pengembang dengan lebih cepat. Urutan langkah pada paradigma *prototype* diilustrasikan dalam gambar 1 di bawah ini.



Gambar 1: Model pengembangan

Untuk membangun layanan *collaborative storage*, ada beberapa infrastruktur yang digunakan, terdiri dari instances Google Compute Engine pada Google Cloud Platform, Engine Nextcloud, docker sebagai container, dan Kubernetes sebagai tools untuk orkestrasi. Masing-masing VM instances memiliki 2 virtual core dan memori 8GB seperti terlihat pada gambar 2.



Gambar 2: Rancangan infrastruktur collaborative research platform

IV. IMPLEMENTASI DAN PENGUJIAN

Langkah dalam implementasi dibangun dalam 2 bagian penting yang tentu saja memiliki tingkat kesulitan tersendiri. Bagian pertama adalah implementasi ini adalah *deploy platform* yang terdiri dari instance, kubernetes dan nextcloud, selanjutnya bagian kedua adalah implementasi *collaborative document* yang merupakan server terpisah dari bagian pertama.

A. Implementasi

1) Instances, Kubernetes dan Nextcloud

Proses implementasi merupakan pemanfaatan *platform collaborative research* untuk keperluan riset Bersama. Implementasi menggunakan *google cloud platform* pada *instance compute engine*, untuk membuat 2 instance tersebut, menggunakan perintah berikut ini pada *console cloud* :

```
export ZONE="us-east-1-c"
export INSTANCE_NAME="nextcloudakakom"
export INSTANCE_TYPE="n1-standard-8"
gcloud compute instances create $INSTANCE_NAME \
  -- zone=$ZONE \
  -- image-project=Ubuntu-18.04 \
```

- machine -type=\$INSTANCE TYPE\
- boot -disk -size =20GB\

Setelah proses *deploy instances*, salah satu VM digunakan untuk membuat layanan nextcloud yang menggunakan infrastruktur Kubernetes, berikut ini beberapa prosesnya:

```
$ sudo apt install docker.io
$ sudo systemctl enable docker
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg
$ sudo apt-add-repository "deb http://apt.kubernetes.io/kubernetes xenial main"
$ sudo apt-get update
$ sudo apt-get install kubelet kubeadm kubectl
$ sudo swapoff -a
$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
$ kubectl taint nodes --all node-role.kubernetes.io/master-
```

Setelah proses Kubernetes, Langkah selanjutnya mempersiapkan database MariaDB ke dalam infrastruktur Kubernetes.

```
$ mkdir nc-deployment
$ cd nc-deployment
$ wget https://raw.githubusercontent.com/andremotz/nextcloud-kubernetes/master/kubernetes-yaml/db-deployment.yaml
$ nano db-deployment.yaml
$ kubectl create -f db-deployment.yaml
$ wget https://raw.githubusercontent.com/andremotz/nextcloud-kubernetes/master/kubernetes-yaml/db-svc.yaml
$ kubectl create -f db-svc.yaml
```

Setelah proses *deploy instances* dan persiapan database selesai, barulah dilakukan *deploy* layanan nextcloud pada kubernetes yang telah disiapkan.

```
$ wget https://raw.githubusercontent.com/andremotz/nextcloud-kubernetes/master/kubernetes-yaml/nc-deployment.yaml
$ nano nc-deployment.yaml
$ kubectl create -f nc-deployment.yaml
$ wget https://raw.githubusercontent.com/andremotz/nextcloud-kubernetes/master/kubernetes-yaml/nc-svc.yaml
$ kubectl create -f nc-svc.yaml
```

Langkah terakhir adalah mempersiapkan proxy untuk layanan nextcloud. Proxy digunakan untuk mengarahkan container docker untuk membantu proses *forwarding* ke https.

```
$ wget https://raw.githubusercontent.com/andremotz/nextcloud-kubernetes/master/kubernetes-yaml/nginx.conf
$ nano nginx.conf
$ wget https://raw.githubusercontent.com/andremotz/nextcloud-kubernetes/master/kubernetes-yaml/proxy-pod.yaml
$ nano proxy-pod.yaml
$ kubectl create -f proxy-pod.yaml
```

2) Collaborative documents

Collabora Online yang digunakan dalam penelitian ini adalah office suite online open-source berbasis

LibreOffice dan self-host. Fitur-fiturnya antara lain:

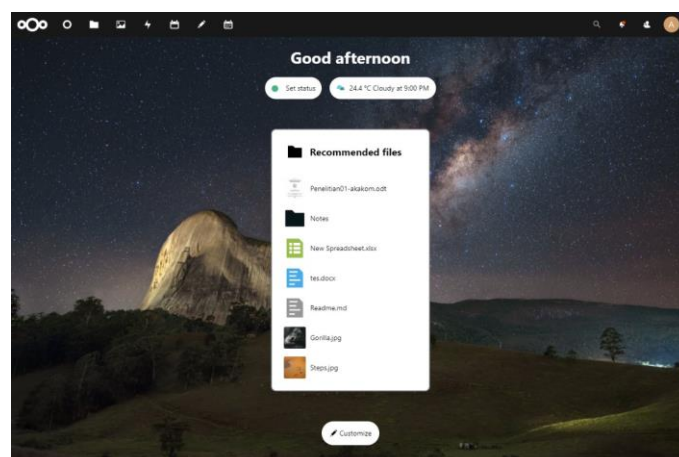
- 1) Pengeditan dasar
- 2) Fidelitas tinggi, rendering WYSIWYG
- 3) Mendukung format dokumen DOC, DOCX, PPT, PPTX, XLS, XLSX, ODF
- 4) Impor dan tampilan Visio, Publisher, dan 100+ lainnya
- 5) Pengeditan Bersama

Collabora adalah kontributor besar untuk proyek LibreOffice. Semua kode Collabora Online pada akhirnya akan dimasukkan ke dalam LibreOffice. Proses pembuatan layanan untuk collaborative ini akan menyatu dengan layanan nextcloud yang sebelumnya dibuat menggunakan Kubernetes. Collabora akan menjadi layanan document collaborative research.

```
$ echo 'deb https://www.collaboraoffice.com/repos/CollaboraOnline/CODE-ubuntu1804 .' | sudo tee /etc/apt/sources.list.d/collabora.list
$ sudo apt-key adv --keyserver keyserver.ubuntu.com
$ sudo apt install
--apt-recv --apt-key 0C54D189F4BA284Dficates
$ sudo apt update
$ sudo apt install loolwsd code brand
$ systemctl status loolwsd
$ sudo loolconfig set ssl.enable false
$ sudo loolconfig set ssl.termination true
$ sudo loolconfig set storage.wopi.host home.akakom.my.id
$ sudo loolconfig set admin.password
$ sudo systemctl restart loolwsd
$ systemctl status loolwsd
$ sudo apt install apache2
$ sudo nano /etc/apache2/sites-available/collabora.conf
$ sudo a2enmod proxy proxy_wstunnel proxy_http
$ sudo a2ensite collabora.conf
$ sudo systemctl restart apache2
$ sudo apt install certbot
$ sudo apt install python3-certbot-apache
$ sudo certbot --apache --agree-tos --redirect --hsts
--staple-ocsp --email m.agung.n@akakom.ac.id
--d collabora.akakom.my.id
```

Dua bagian dari implementasi telah dilakukan, langkah selanjutnya adalah melakukan setting minor pada nextcloud agar dapat berjalan dengan baik dan sesuai rencana awal.

Dalam proses ini pun dilakukan akses ke *dashboard* dan penggunaan umum untuk *sharing resource* dan kolaborasi dokumen. Halaman dashboard seperti terlihat pada 3 adalah halaman utama tempat peneliti dapat memulai aktivitas, disini dapat melakukan akses ke halaman file manager untuk akses ke file management bersama. Selain itu juga dapat mengakses beberapa plugin seperti git untuk membuat akses kolaborasi *source code* bersama.



Gambar 3: Halaman dashboard

Pada halaman file *manager research collaborative* bermanfaat untuk melihat beberapa kebutuhan file dan document yang digunakan untuk proses kolaborasi. Juga akses ke git tempat berbagi *source code* atau kode program dari riset yang dilakukan oleh peneliti. Platform ini juga memberikan kemudahan untuk para kolaborator bersama untuk menyelesaikan dokumen seperti terlihat pada gambar 4. Secara umum fitur ini sudah seperti layaknya menggunakan aplikasi word di desktop. Perbedaannya para kolaborator dapat menulis bersama dalam satu dokumen yang sama, sehingga dapat memudahkan *sharing* penulisan.



Gambar 4: Document collaboration

B. Pengujian

Proses pengujian dilakukan dengan menguji *Performance speed*, dan Performa berbasis *user request*. Tujuan dari pengujian ini adalah untuk mengetahui seberapa baik performa dari server Nextcloud yang sudah dibangun dan mengukur tingkat *availability* pada layanan *collaborative research*.

V. HASIL DAN PEMBAHASAN

A. Performance speed test

Pada pengujian performa penulis melakukan pengujian menggunakan GTmetrix untuk mengetahui nilai ataupun skor performa seperti *page speed score*, *yslow score* dan mengetahui *page details* seperti *fully loaded time* pada halaman awan yaitu <https://home.akakom.my.id>. Hasil pengujian dapat dilihat pada gambar 5



Gambar 5: Performance Speed Test

Web *collaborative research* memiliki peringkat performa kecepatan akses A, dimana secara performance memiliki tingkat kecepatan akses 90% dan struktur akses sebesar 92%. Adapun rincian secara detil adalah seperti terlihat pada tabel I:

Berdasarkan data metrik dari GT Metrix, web collaborative research masuk dalam kategori good atau baik. Dan memiliki kecepatan akses rata-rata dibawah 2s.

B. Performance test berbasis user request

Pengujian performa ini digunakan untuk mengukur tingkat *availability* pada layanan *collaborative research* atau dikenal dengan *load testing*. Tujuannya adalah untuk mengetahui maksimum jumlah yang dapat mengakses layanan ini.



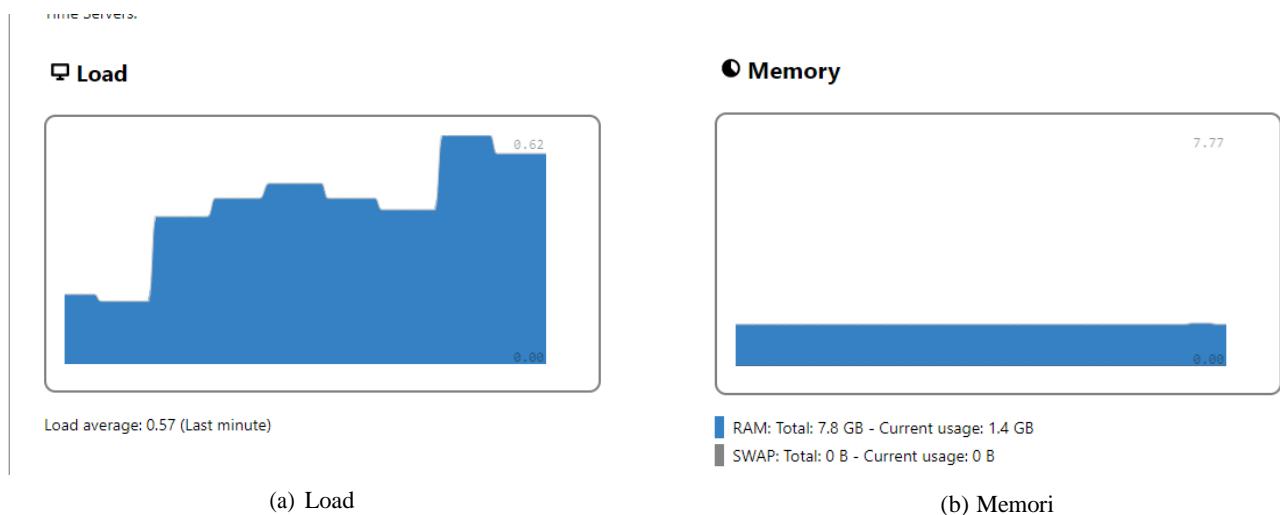
Gambar 6: Statistik load testing dari Blazemeter

Dengan penggunaan virtual machine 2 core dan memori 8 GB, dapat dihasilkan beberapa data testing. Pengujian dilakukan dengan 20 user, karna pengkhususan dari penelitian ini adalah *collaborative research* dimana biasanya dalam 1 project penelitian terdiri dari 1 - 5 peneliti yang berkolaborasi, sehingga jumlah 20 pengguna dianggap telah memenuhi.

Dengan menggunakan 20 pengguna *response time* dari layanan ini berjalan dibawah rata-rata load tinggi, yaitu di peak tertinggi adalah diatas 30000 ms. Sementara layanan ini juga memiliki *connect time* yang baik antara 0 – 5 ms. Dan pada level hit/s error memiliki rata-rata yang hampir sama dengan *connect time*. Pada tahapan ini layanan dapat digunakan pada pengguna maksimum 20 orang.

TABEL I: METRIK PERFORMANCE KECEPATAN AKSES

Metrics	Nilai
First Contentful Paint	1.3s
Time to Interactive	1.4s
Speed Index	1.3s
Total Blocking Time	43ms
Cumulative Layout Shift	0.01
Redirect Duration	385ms
Connection Duration	1ms
Backend Duration	114ms
Time to First Byte (TTFB)	0.5s
DOM Interactive Time	0.7s
DOM Content Loaded Time	1.4s
Onload Time	1.5s



Gambar 7: Statistik penggunaan resource

Dalam proses pengujian *load testing* dengan blazemeter dengan membandingkan hasil statistik sumber seperti terlihat pada gambar 7, menunjukkan bahwa ketika terjadi jumlah beban yang tinggi dalam akses ke layanan ini tidak mempengaruhi dari sisi penggunaan prosesor dan memori. Penggunaan memori hanya 1.4 GB dari total 7.8 GB, yang masih mencukupi untuk proses dari layanan ini.

VI. SIMPULAN

Kubernetes dan Nextcloud merupakan *platform* yang dapat menjadi solusi dalam membangun sebuah *cloud storage* untuk *collaboratin research*. Pengujian yang dilakukan membuktikan bahwa *platform* ini mendapatkan nilai yang baik dengan skala A pada pengujian *Performance access speed*. Dengan penggunaan 20 pengguna pada pengujian *performance user request*, dan dapat berjalan pada *peak* dengan connect time 0 - 5ms dan level bit/s error menunjukkan nilai rata-rata yang tidak jauh berbeda.

Penurunan performa yang dapat terjadi dapat disolusikan lebih optimal dengan menggunakan model *scaling* pada container dengan kubernetes, dapat pula *scaling* ini digabungkan dengan metode *scaling* dari layanan *cloud google* dengan kombinasi penggunaan proxy, *load balancing*, dan *scaling* fitur dari kubernetes dan *google cloud platform*. Pada penelitian ini NextCloud hanya diimplementasikan dalam *private cloud* pada teknologi Kubernetes dan tidak dibahas masalah keamanan yang mungkin terjadi pada implementasinya dan hanya pada 20 pengguna saja.

UCAPAN TERIMA KASIH

Ucapan terima kasih kami harurkan kepada LPPM STMIK Akakom Yogyakarta atas dukungan dana agar penelitian ini terlaksananya dengan baik dan kepada seluruh sivitas akademika STMIK Akakom Yogyakarta, atas kerjasama dan dukungannya.

PUSTAKA

- [1] B. A. Ward and W. J. Tikunoff, "Collaborative Research," *Natl. Invit. Conf. "Reseach Teach. Implic. Pract.*, vol. 53, no. 9, pp. 1689–1699, 1982.
- [2] X. Liu, "Full-Text Citation Analysis : A New Method to Enhance," *J. Am. Soc. Inf. Sci. Technol.*, vol. 64, no. July, pp. 1852–1863, 2013. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/asi.22883/abstract>
- [3] M. Taufer, P. Teller, A. Kerstens, and R. Romero, "Collaborative Research Tools for Students, Staff, and Faculty," in *Retrieved 10th Novemb.*, no. January, 2007, pp. 1–6. [Online]. Available: <http://sunconference.utep.edu/sunhome/2006/docs/proceedings/CollaborativeResearchTools.pdf>
- [4] T. N. Developers, *Nextcloud User Manual The Nextcloud developers*. Nextcloud, 2021.
- [5] F. Karlitschek and N. Mache, "Nextcloud Solution Architecture Bring data back under control of IT Overview of the Nextcloud Architecture," p. 17, 2017.
- [6] "Docs & Guides - ownCloud." [Online]. Available: <https://owncloud.com/docs-guides/>
- [7] "Database Configuration — Nextcloud 12 Server Administration Manual 12 documentation." [Online]. Available: https://docs.nextcloud.com/server/12/admin_manual/configuration_database/linux_database_configuration.html
- [8] L. P. De wi, A. Noertjahyana, H. N. I alit, and K. Yedutun, "Server Scalability Using Kubernetes," *Eng. Sci. Int. Conf.*, pp. 1–4, 2019.
- [9] M. A. Nugroho and R. Kartadie, "ANALISIS KINERJA PENERAPAN CONTAINER UNTUK LOAD BALANCING WEB SERVER PADA RASPBERRY PI," *JIPI*, vol. 01, no. 02, pp. 7–15, 2016.
- [10] S. Kariyattin, S. Marru, and M. Pierce, "Evaluating NextCloud as a file storage for apache airavata," *ACM Int. Conf. Proceeding Ser.*, pp. 1–4, 2018.
- [11] V. Medel, R. Tolosana-Calasanz, J. Á. Bañares, U. Arronategui, and O. F. Rana, "Characterising resource management performance in Kubernetes," *Comput. Electr. Eng.*, vol. 68, no. March, pp. 286–297, 2018. [Online]. Available: <https://doi.org/10.1016/j.compeleceng.2018.03.041>
- [12] A. R. Putri, R. Munadi, and R. M. Negara, "Performance analysis of multi services on container Docker, LXC, and LXD," *Bull. Electr. Eng. Informatics*, vol. 9, no. 5, pp. 2008–2011, 2020.
- [13] M. Beaudouin-lafon and W. E. Mackay, "Chapter 52 Prototyping Tools and Techniques," in *Prototype Dev. Tools*, 2000, pp. 1–41.